汇编语言程序设计 Assembly Language Programming

第一章基础知识

§1数的表示

- *数制
- *数制之间的转换
- * 计算机中数的表示
- **❖ BCD码**
- *字符编码

1.1 数 制

- → 十进制: 基数为10, 逢十进一
 - $12.34D = 1 \times 10^{1} + 2 \times 10^{0} + 3 \times 10^{-1} + 4 \times 10^{-2}$
- **→ 二进制**: 基数为2, 逢二进一

$$1101B = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 = 13_{10}$$

→ 十六进制:基数为16,逢十六进一

) 1

7

$$= 9 \times 16^3 + 1 \times 16^2 + 8 \times 16^1 + 7 \times 16^0$$

→ 八进制:基数为8,逢八进一

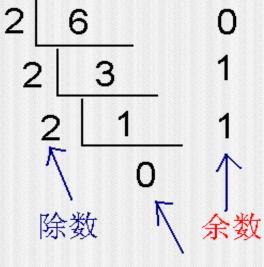
1.2 数制之间的转换

二进制→十进制

1011B = 11D

除法:

❖ 整数部分除2; 小 数部分乘2。



商/被除数

步骤演化:					
2	6	_ 0			
① 6	3 ÷ 2,	商2余0			
2 [6	_ 0			
2	3	_ 1			
② 3	1 ÷ 2,	商1余1			
	-				

▶小数可能不能用二进制来表示完全,如0.6

▶二进制 → 十六进制

- : A19CH = 1010,0001,1001,1100B
- ▶ 4位二进制对应一位16进制

→十六进制 —十进制

BF3CH = 11×16³ + 15×16² + 3×16¹ + 12×16⁰ 降幂法 除法

♥常用数

0—00H 128—80H 255—FFH 256—100H 32767—7FFFH 65535—FFFFH

1.3 计算机中数的表示

- 肯定是用二进制
- ▶无符号数:直接用"二进制"来表示。
- ▼有符号数:原码、反码、补码。
- ♥ 浮点数: (尾数) 规格化+(指数) 移码减一

补码

→定义

- ❖最高有效位为符号位: 0—正; 1—负
- ❖ 正数的补码为它本身(二进制值);负数的补码由 2ⁿ-|x|求得。n为机器的字长。
- ❖注: 2ⁿ-|x| 等价于 对|x|取反+1
- ♦ [46]
 ♦ [46]
 ♦ 0010 1110
- * [-46] * [-46] * [-46] = 1,0000,0000 - 0010 1110 = 11010010
- * [-46]; |-46| = 46 = 0010 1110 → 11010010

补码的特性

> 求补特性

- ❖求补:取反加1
- ❖补码为11001100, 十进制数为?

→加减

- ❖ 这个特性使得对于补码数或者一般的二进制数对加减规则一样。不可用于乘除运算。
- ❖ [x-y]_补 = [x]_补+ [-y]_补: 减法可用加法实现。

n位二进制补码的表数范围

SM	十进制	二进制	十六进制	十进制	十六进制
		n=8		个带符号数在不=	\
	+127	0111 1111	7月 同	位数+32757二进	\ 7FFF
	+126	0111 1110	7萬 制	补码表270能是	/ 7FFE
	•••	•••		不同的	/
	+2	0000 0010	02	12	0002
	+1	0000 0001	01	+1	0001
	0	0000 0000	00 V	0	0000
	-1	1111 1111	FF	-1	FFFF
	-2	1111 1110	FE	-2	FFFE
	•••	• • •	•••	•••	•••
	-126	1000 0010	82	-32766	8002
	-127	1000 0001	81	-32767	8001
	-128	1000 0000	80	-32768	8000

数的范围

- → 无符号数: 8位(0---255) 16位 (0---65535)
- ♥有符号数:

8位 (-128---127) 16位 (-32768---32767)

♥n位补码表示数范围: - 2ⁿ⁻¹ ≤ N ≤ 2ⁿ⁻¹-1

补码:C语言

- →其实C语言中unsigned和signed两个关键字就是来区别无符号数和有符号数,缺省的是signed类型,也就是有符号数。
- ♥实例:
 - \diamond char i = -1;
 - \star short j = -1;
 - \star int k = -1

- ▶ 浮点数在计算机中指的是带有小数的数值,通常为32位或64位,分别被叫作单精度(float)和双精度(double)。
- ▶ 浮点数由一个整数或定点数(即尾数)乘以某个基数(计算机中通常是2)的整数次幂得到:

$$a = m * b^e$$

▶ 浮点数表示方法类似于基数为 10 的科学记数法: 3.14*10³、1.01*2⁸;

▶ 浮点数的存储 (a = m * b^e):

```
sign exponent fraction

e+f f 0
```

- ▶ 符号位: 尾数的符号, 1为负, 0为正;
- ▶ 尾数: 即m规格化的值;
- ▶ 指数: 即指数e的移码(即将其补码的符号位取反)-1;

- > 尾数必须规格化:
 - ➤ IEEE 754 标准: 尾数必须是1.xxx形式,这样小数形式 唯一确定。
 - $> (5.0)_{10} = (101.0)_{2} = (1.01*2^{2})_{2}$
 - ➤ 存储时只存储.xxx部分,这样多保持一位有效数字,上 例存储01,即尾数的数值;
- ▶示例:
 - $(0.15625)_{10} = (0.00101)_{2} = (1.01*2^{-3})_{2}$
 - -3的补码是1111 1101, 其移码是0111 1101

▶ 浮点数形式:

类型	位数	数符	阶码	尾数
单精度浮点数(float)	32 位	1位	8位	23 位
双精度浮点数 (double)	64 位	1位	10 位	53 位
临时浮点数	80 位	1位	15 位	64 位

1.4 BCD (Binary-Coded Data) 码

- ♥Why? 十进制用起来很方便。
- ▶二进制编码的十进制:
 - ❖ Packed BCD: 用4位二进制表示一个十进制数码
 - 0000-----00001-----1
 - 0001,0010,0011,0100 = 1234
 - ❖ Unpacked BCD: 用8位二进制表示一个十进制数码
 - ****0000----0 ****0001----1
 - *****,0001,****,0010, ****,0011, ****,0100 = 1234

- ♥如何表示一个长度为100的十进制数?
 - ❖用长度为100的数组来表示
- ▶如何实现两个长度为100的数的加法?
 - *用数组加
- ▶如何实现两个长度为100的数的乘法?
- →如何实现长度为200的数除以长度为100的数?

公钥体制和数字签名的基本原理

- →P和Q为长度为100左右的质数,P*Q=N较 为容易
- ▶由N求P和Q特别难
- →公钥:N对外公开,用来加密,与PQ有关的数作为私钥,解密
- ▶数字签名: 用私钥对要签名内容的散列值加密, 用N解密验证。

1.5 字符编码

- ♥ ASCII: 英文, 7 bits
- ♥ 常用字符的ASCII码。
 - ❖ 数字'0'~'9': 30H~39H
 - ❖ 字母'A'~'Z': 41H~5AH
 - ❖ 字母'a'~'z': 61H~7AH
 - ❖ 空格: 20H
 - ❖ 回车CR: 0DH
 - ❖ 换行LF: 0AH
 - ❖ 空字符: 0
 - ❖ 注意回车与换行的差别: CR用来控制光标回到当前行的最左端; LF用来移动光标到下一行, 而所在列不变。

1.5 字符编码

♥UNICODE: 16位二进制 = 65536

汉字: 20902个

♥UTF-8: 用4字节表示, 2³² = 42亿....